BHARATI VIDYAPEETH

(Deemed to be University), Pune

'A++' Accreditation (Third Cycle) by 'NAAC' in 2024

Category-I Deemed to be University

Graded by UGC

‘A’ Grade University Status by MHRD Govt. of India.

FACULTY OF MANAGEMENT STUDIES

MASTER OF COMPUTER APPLICATION

FRAMED AS PER NATIONAL EDUCATION POLICY (NEP 2020)

SYLLABUS

Applicable with effect from 2026-27

Content

Bharati Vidyapeeth (Deemed To Be University), Pune
Faculty of Management Studies
(Board of Studies in Computer Applications and System Studies)
Master of Computer Application (2026 Course)
(Under NEP 2020)
Effective from Academic Year 2026-27

I. INTRODUCTION:

The MCA Program is a full time 95 credits programme offered by Bharati Vidyapeeth (Deemed to be University), Pune and is conducted in regular and distance mode at its Management Institutes in Pune, Karad, Kolhapur, Sangli, and Solapur. This programme is also conducted in online mode at CDOE under BV(DU). All the five institutes have excellent faculties, laboratories, library, and other facilities to provide proper learning environment. The University is reaccredited by NAAC with an 'A++' grade (3rd cycle). The expectations and requirements of the software industry, immediately and in the near future, are visualized while designing the MCA programme. This effort is reflected in the Vision and Mission statements of the MCA programme. Of course, the statements also embody the spirit of the vision of Late Dr. Patangraoji Kadam, the Founder of Bharati Vidyapeeth and Chancellor, Bharati Vidyapeeth (Deemed to be University) which is to usher in "Social Transformation through Dynamic Education."

II. VISION

Achieve excellence in Computer Applications with respect to teaching, learning and research to meet the growing needs of the industry and society.

III. MISSION

- Promote outcome-based learning strategies in-order to meet global industry standards.
- Encourage innovations and problem-solving capabilities in students and faculty.
- Cultivate collaborative research in both, students and faculty members through industry interactions and collaborations.
- Enhance entrepreneurship skills among students.

IV. PROGRAMME UNIQUE FEATURES

Keeping the view of National Education Policy, MCA Programme is designed with following features

- Students can choose any of the elective group through which he/she will be trained in specialized area for better career.
- Internship project provides a platform which gives acquaintance for solving IT problems.

V. PROGRAMME OBJECTIVES

1: To build a strong foundation for students to become proficient in all academic concepts and technical skills necessary to become an IT Professional.

2: To provide a conducive environment for designing, implementing and testing various software applications through Software Development.

3: To keep the students and faculty abreast with the emerging technologies in the field of computer applications.

4: To bring professionalism amongst the students and promote holistic development.

5: To involve students in sustainable IT practices and community services.

VI. PROGRAMME OUTCOMES (PO)

PO1: Computational Knowledge: Apply knowledge of computing fundamentals, mathematics and given domain to design appropriate models for a given problem and/or requirements.

PO2: Problem Analysis: Apply fundamental knowledge of software engineering and various systems domain in order to analyze, identify, formulate and provide the solution to given problem.

PO3: Software Design and Development: Design and evaluate solutions, systems, modules and processes for specified set of needs with appropriate consideration of societal values and industry expectations.

PO4: Modern Computing Tools Usage: Use of modern tools for delivering milestones like problem analysis, design, development, testing and deployment.

PO5: Research, Innovation & Entrepreneurship Computing problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. Faster innovative & Entrepreneurial thinking to develop solutions that address industrial, business and societal challenges

PO6: Professional Ethics: Learn and inculcate professional ethics, cyber regulations, professional responsibilities and norms of professional computing world.

PO7: Communication Efficacy & Lifelong Learning: Communicate effectively with diverse stakeholders and engage in continuous learning to adapt to emerging technologies, evolving professional practices, and global technological advancements.

VII. PROGRAMME SPECIFIC OUTCOMES

Programme Specific Objectives (PSOs) define the distinctive competencies that graduates of the MCA programme at IMED are expected to demonstrate beyond the general programme outcomes.

- **PSO 1: Advanced Computing & Software Development:** Apply modern computing principles, software development methodologies, and emerging technologies to design, develop, deploy, and maintain secure, scalable, and efficient software solutions for diverse application domains.
- **PSO 2: Intelligent Digital Solutions & Innovation:** Leverage Artificial Intelligence, Data Science, Cloud Computing, Cyber security, and related technologies to develop innovative, sustainable, and socially responsible digital solutions while demonstrating entrepreneurial thinking and professional ethics.

VIII. ELIGIBILITY CRITERIA :

Admission to the programme is open to any Graduate (10+2+3) of any recognized University satisfying the following conditions.

1. Passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree. OR Passed B.Sc./ B.Com./ B.A. with additional bridge Courses (Bridge Course I/ Bridge Course II)as per the norms of the University.

OR

Passed any graduation degree (e.g. BE/ BTech/ BSc/BCom/BA/B.Voc/ etc) preferably with mathematics at 10+2 level or at Graduation Level

2. The candidate should have secured at least 50% marks (45% for SC/ST) in aggregate at graduate level university examination.
3. For students having no Mathematics background compulsory bridge course framed by the Bharati Vidyapeeth (Deemed to be University) and additional bridge course related to computer subjects should be completed
4. The candidate studying in final year of Bachelor's degree may also apply. Admission of such candidates will remain provisional until submission of final result certificates in original.
5. Subject to the above conditions, the final admission of final admission is based solely on –

a. The merit at All India Entrance Test conducted by Bharati Vidyapeeth (Deemed to be University), Pune.
b. Submission of Migration Certificate, Transference Certificate, anti-ragging affidavit etc.

IX. DURATION OF THE PROGRAMME

The duration of this programme is two years divided into four semesters or a minimum of 95 credits whichever is later.

The medium of instruction and examination will be only English.

X. MOOC Policy :

In alignment with global educational standards and the vision of Bharati Vidyapeeth (Deemed to be University), Pune to deliver dynamic education, Massive Open Online Courses (MOOCs) are integrated as a compulsory component of the curriculum. This policy is designed to cultivate a culture of **lifelong learning**, self-discipline, and global competency among students. Every student is mandatorily required to successfully complete two (2) independent MOOCs during their academic journey. These courses run parallel to regular classroom studies to foster self-paced study habits essential for continuous personal and professional development.

The Institute may recommend or prescribe suitable online learning platforms for completion of MOOCs/online certification courses from time to time, based on the relevance of the course, academic requirements, and regulatory guidelines. Students shall complete the prescribed MOOCs from the platforms approved by the Institute. Indicative platforms include, but are not limited to:

1. SWAYAM
2. NPTEL
3. Coursera
4. edX
5. Google Cloud Skills Boost
6. Microsoft Learn
7. AWS Skill Builder
8. Oracle University
9. Cisco Networking Academy
10. IBM SkillsBuild
11. Salesforce Trailhead
12. Red Hat Training
13. NVIDIA Deep Learning Institute

The above list is indicative and not exhaustive. The Institute reserves the right to add, remove, or modify the approved platforms and prescribe specific courses as per the curriculum requirements and emerging industry trends.

Important Note:

- Students can complete MOOCs anytime during the valid tenure of the programme
- Students have to submit completion Certificate of MOOCs. Unless certificate of two (2) MOOCs submitted, Fourth Semester Marksheet will not be issued.

XI. SCHEME OF EXAMINATION:

For some courses there is Internal Assessment (IA) conducted by the respective institutes continuously as well as a University Examination (UE) at the End-of-the Term. UE will be conducted out of 50 marks and IA will be conducted for 50 marks then these are converted to grade points and grades as per the Table I. For courses having only Continuous Assessment (CA) the respective institutes will evaluate the students in varieties of ways during the term for a total of specified marks. Then the marks will be converted to grade points and grades using the Table I.

XII. STANDARD OF PASSING:

For all courses, both UE and IA constitute separate heads of passing (HoP). In order to pass in such courses and to earn the assigned credits, the student/learner must obtain a minimum grade point of 5.0 (40% marks) at UE and also a minimum grade point of 5.0 (40% marks) at IA.

If Student fails in IA, the learner passes in the course provided, he/she obtains a minimum 25% marks in IA and GPA for the course is at least 6.0 (50% in aggregate). The GPA for a course will be calculated only if the learner passes at UE.

A student who fails at UE in a course has to reappear only at UE as backlog candidate and clear the Head of Passing. Similarly, a student who fails in a course at IA he has to reappear only at IA as backlog candidate and clear the Head of Passing to secure the GPA required for passing. The 10 point Grades and Grade Points assign according to the following table

Range of Marks (Out of 100)	Grade	Grade Point
90≤Marks≤100	O	10
80≤Marks<90	A+	9
70≤Marks<80	A	8
60≤Marks<70	B+	7
50≤Marks<60	B	6
40≤Marks<50	C	5
Marks < 40	D	0

Table I : Grade Points and Grades

The performance at UE and IA will be combined to obtain GPA (Grade Point Average) for the course.The weights for performance at UE and IA shall be 50% and 50% respectively.

GPA is calculated by adding the UE marks out of 50 and IA marks out of 50.The total marks out of 100 are converted to grade point, which will be the GPA.

Rules of ATKT

For course upto four semesters, a student is allowed to carry any number of Backlogs of a prescribed course in Sem-I, II, III provided he appears and have backlogs.

A student can appear for any four continuous semesters in an examination season including the regular semester, provided the student has appeared and have backlogs for other three semesters.

Formula to calculate Grade Points (GP)

Suppose that "Max" is the maximum marks assigned for an examination or evaluation, based on which GP will be computed. In order to determine the GP, Set x = Max/10 (since we have adopted 10 point system). Then GP is calculated by the following formulas

Range of Marks	**Formula for the Grade Point**
8x ≤ Marks≤10x	10
5.5x ≤ Marks≤8x	Truncate (M/x) +2
4x ≤ Marks≤5.5x	Truncate (M/x) +1

Two kinds of performance indicators, namely the Semester Grade Point Average (SGPA) and the Cumulative Grade Point Average (CGPA) shall be computed at the end of each term. The SGPAmeasures the cumulative performance of a learner in all the courses in a particular semester, while the CGPA measures the cumulative performance in all the courses since his/her enrollment. The CGPA of learner when he /she completes the programme is the final result of the learner.

The SGPA is calculated by the formula

$$SGPA = \frac{\sum Ck * GPk}{\sum Ck}$$

where, Ck is the Credit value assigned to a course and GPk is the GPA obtained by the learner in the course. In the above, the sum is taken over all the courses that the learner has undertaken for the study during the Semester, including those in which he/she might have failed or those for which he/sheremained absent. **The SGPA shall be calculated up to two decimal place accuracy.**

The CGPA is calculated by the following formula

$$CGPA = \frac{\Sigma C_k * GP_k}{\Sigma C_k}$$

where, Ck is the Credit value assigned to a course and GPk is the GP obtained by the learner in the course. In the above, the sum is taken over all the courses that the learner has undertaken for the studyfrom the time of his/her enrollment and also during the semester for which CGPA is calculated. The CGPA shall be calculated up to two decimal place accuracy.

The formula to compute equivalent percentage marks for specified CGPA:

CGPA Range	Formula for Percentage (%)	Applicable Condition
5.00 ≤ CGPA < 6.00	Percentage = (10 × CGPA) − 10	For CGPA from 5.00 (inclusive) to 6.00 (exclusive)
6.00 ≤ CGPA < 8.00	Percentage = (5 × CGPA) + 20	For CGPA from 6.00 (inclusive) to 8.00 (exclusive)
8.00 ≤ CGPA < 9.00	Percentage = (10 × CGPA) − 20	For CGPA from 8.00 (inclusive) to 9.00 (exclusive)
9.00 ≤ CGPA < 9.50	Percentage = (20 × CGPA) − 110	For CGPA from 9.00 (inclusive) to 9.50 (exclusive)
9.50 ≤ CGPA ≤ 10.00	Percentage = (40 × CGPA) − 300	For CGPA from 9.50 (inclusive) to 10.00 (inclusive)

XIII. Award of Honors :

A student who has completed the minimum credits specified for the programme shall be declared to have passed in the programme. The final result will be in terms of letter grade only and is based on the CGPA of all courses studied and passed. The criteria for the award of honours are given below.

Range of CGPA	Final Grade	Performance Descriptor	Equivalent Range of Marks (%)
9.5≤CGPA ≤10	O	Outstanding	80≤Marks≤100
9.0≤CGPA ≤9.49	A+	Excellent	70≤Marks≤80
8.0≤CGPA ≤8.99	A	Very Good	60≤Marks≤70
7.0≤CGPA ≤7.99	B+	Good	55≤Marks≤60
6.0≤CGPA ≤6.99	B	Average	50≤Marks≤55
5.0≤CGPA ≤5.99	C	Satisfactory	40≤Marks≤50
CGPA below 5.0	F	Fail	Marks below 40

XIV. Internal Assessment scheme

Internal Assessment (IA) shall be conducted throughout the semester using appropriate assessment methods aligned with the Course Outcomes (COs). The assessment methods are indicative and may be adapted by the course instructor based on the nature of the course and learning objectives. The following guidelines shall be followed for the conduct and implementation of the Internal Assessment (IA) Scheme.

a. Internal Assessment Scheme (50 Marks)

Assessment Component	Indicative Assessment Methods	Marks
Assessment I: Individual Assessment	Written Test, Practical/Coding Test, Problem Solving, Design Exercise, Technical Report, Case Analysis, Open Book Test, Reflection Report	**15**
Assessment II: Collaborative Assessment	Project, Case Study, Group Presentation, Seminar, Prototype Demonstration, Group Discussion	**15**
Assessment III: Continuous Assessment	Quiz, Tutorial, Practical/Laboratory Exercise, Viva Voce, Short Test, Learning Portfolio	**10**
Assessment IV: Professional Engagement & Lifelong Learning	Technical Portfolio, Workshop/Seminar Report, MOOC/Certification Completion, Reflective Learning Report, Professional Conduct Evaluation	**10**
	Total	50

b. Internal Assessment Scheme (100 Marks)

Assessment Component	Indicative Assessment Methods	Marks
Assessment I: Individual Assessment	Written Test, Practical/Coding Test, Problem Solving, Design Exercise, Technical Report, Case Analysis	**25**
Assessment II: Collaborative Assessment	Project, Industry Case Study, Group Presentation, Prototype Demonstration, Research Project	**25**
Assessment III: Continuous Assessment	Quiz, Tutorial, Practical/Laboratory Exercise, Viva Voce, Short Test, Learning Portfolio	**20**
Assessment IV: Professional Engagement & Lifelong Learning	Technical Portfolio, Workshop/Seminar Report, MOOC/Certification Completion, Reflective Learning Report, Professional Conduct Evaluation	**15**
Assessment V: Comprehensive Assessment (where applicable)	Project Demonstration, Portfolio Evaluation, Practical Demonstration, Capstone Presentation, Comprehensive Viva Voce	**15**
	Total	100

Indicative Internal Assessment Activities for Different Categories of Courses

The assessment methods presented under the Internal Assessment components are general indicative methods applicable across the programme. As the nature and learning outcomes of courses vary, the actual assessment activities should be selected from the relevant course category-wise assessment activities provided in the following tables. The course instructor may adopt one or more appropriate assessment methods aligned with the Course Outcomes (COs), subject requirements, and Outcome-Based Education (OBE) principles. The activities listed are indicative and not exhaustive, and the Institute may revise or prescribe additional assessment methods, evaluation rubrics, digital platforms, software tools, and learning resources from time to time

Programming, Computing & Laboratory Courses

Assessment Component	Indicative Assessment Methods
Assessment I: Individual Assessment	Coding Test, Debugging Exercise, Algorithm Design, SQL Query Writing, Code Optimization, Program Development, Technical Report, Technology Review
Assessment II: Collaborative Assessment	Software Development Project, Web/Mobile Application Development, Database Design Project, API Integration Project, GitHub Collaborative Project, Software Prototype Development, Hackathon Challenge
Assessment III: Continuous Assessment	Laboratory Exercises, Weekly Coding Exercises, Practical Quiz, Code Review, Programming Viva, Unit Test
Assessment IV: Professional Engagement & Lifelong Learning	Technical Portfolio, MOOC/Certification Completion, Coding Competition Performance, Workshop/Seminar Report, Reflective Learning Report

Mathematics, Statistics & Analytical Courses

Assessment Component	Indicative Assessment Methods
Assessment I: Individual Assessment	Written Test, Numerical Problem Solving, Mathematical Modelling, Data Interpretation, Proof Writing, Analytical Report
Assessment II: Collaborative Assessment	Mathematical Modelling Project, Group Problem Solving, Data Analysis Project, Case-based Numerical Analysis, Seminar Presentation
Assessment III: Continuous Assessment	Tutorial Sheets, Class Tests, Problem-solving Exercises, Practical Quiz
Assessment IV: Professional Engagement & Lifelong Learning	Mathematical Portfolio, MOOC/Certification Completion, Seminar Report, Reflective Learning Report

Database, Networking, Software Engineering & Information Systems Courses

Assessment Component	Indicative Assessment Methods
Assessment I: Individual Assessment	Requirement Analysis, UML Design, SQL Tasks, Network Configuration Exercise, System Design Report, Technical Documentation
Assessment II: Collaborative Assessment	Software Engineering Project, System Design Project, Network Simulation Project, Industry Case Study, Group Technical Presentation
Assessment III: Continuous Assessment	Laboratory Exercises, Configuration Tasks, Design Review, Practical Quiz, Viva Voce
Assessment IV: Professional Engagement & Lifelong Learning	Technical Portfolio, Workshop/Seminar Report, MOOC/Certification Completion, Reflective Learning Report

Artificial Intelligence, Data Science & Emerging Technologies

Assessment Component	Indicative Assessment Methods
Assessment I: Individual Assessment	Data Analysis using Python/R, AI/ML Model Development, Prompt Engineering Task, Data Visualization, Research Review, Technology Review
Assessment II: Collaborative Assessment	AI/ML Project, Data Analytics Project, Cloud-based Application, Research Project, Prototype Development, Group Technical Presentation
Assessment III: Continuous Assessment	Coding Exercises, Laboratory Exercises, Practical Quiz, Model Evaluation Exercise, Viva Voce
Assessment IV: Professional Engagement & Lifelong Learning	Technical Portfolio, Hackathon Performance, Workshop/Seminar Report, MOOC/Certification Completion, Reflective Learning Report

Soft Skills, Communication, Management & Value-Added Courses

Assessment Component	Indicative Assessment Methods
Assessment I: Individual Assessment	Mock Interview, Resume/CV Evaluation, Technical Writing Exercise, Reflection Report, Essay Writing, Elevator Pitch, Individual Presentation, Case Analysis
Assessment II: Collaborative Assessment	Group Discussion, Debate, Role Play, Panel Discussion, Team Presentation, Business Simulation, Community Engagement Project, Case Study Presentation
Assessment III: Continuous Assessment	Short Quiz, Communication Exercise, Scenario-based Problem Solving, Reflective Journal, Learning Portfolio, Peer Assessment
Assessment IV: Professional Engagement & Lifelong Learning	Professional Conduct Evaluation, Leadership & Teamwork Evaluation, Workshop/Seminar Report, E-Portfolio Evaluation, MOOC/Certification Completion

Project, Internship, Seminar & Capstone Courses

Assessment Component	Indicative Assessment Methods
Assessment I: Individual Assessment	Project Proposal, Literature Review, Progress Report, Technical Documentation, Project Diary, Individual Presentation
Assessment II: Collaborative Assessment	Software/System Development Project, Industry Project, Prototype Development, Poster Presentation, Group Demonstration
Assessment III: Continuous Assessment	Milestone Review, Progress Seminar, Supervisor Evaluation, Logbook Evaluation, Project Viva
Assessment IV: Professional Engagement & Lifelong Learning	Technical Portfolio, Reflective Portfolio, Industry Mentor Evaluation, Project Management Report, Workshop/Seminar Report

General Guidelines for Internal Assessment

1. Internal Assessment shall be based on multiple assessment methods to evaluate knowledge, skills, application, communication, collaboration, and professional attributes.
2. The assessment methods listed above are **indicative and not exhaustive**. The course instructor may select one or more appropriate assessment methods depending on the nature of the course and the Course Outcomes (COs).
3. Assessment I should primarily evaluate individual learning and application of course concepts.
4. **Assessment II** should assess **collaborative learning, teamwork, and application of knowledge**.
5. **Assessment III** should support **continuous learning and formative assessment** throughout the semester.

6. **Professional Engagement & Continuous Learning** shall be evaluated using documented evidence such as attendance, participation, laboratory engagement, teamwork, professional conduct, communication, timely submission of assignments, self-learning initiatives, MOOCs, workshops, technical events, coding competitions, GitHub/e-portfolio, or other relevant activities.

XV. Question Paper Patterns for University Examination:

Title of the Course

Day: **Total Marks: 50**

Date: **Time: 03 Hours**

Instructions:

1. **Section I –Attempt any two questions. Each question carries 05 Marks.**
2. **Section II –Attempt any two questions. Each question carries 10 Marks.**
3. **Section III –Question 7 is compulsory and attempt any one question from the remaining two. Each question carries 10 Marks.**
4. **All three sections should be attempted in same answer sheet.**

SECTION – I			
Question	**Marks**	**CO**	**BL**
Q.1 Write Short Notes on ANY Two **a)** **b)** **c)**	**(5 marks)**		**Remember**
Q.2 ……	**(5 marks)**		**Understand**
Q.3 ……	**(5 marks)**		**Understand**
SECTION – II			
Q.4 ……	**(10 marks)**		**Apply**
Q.5 ……	**(10 marks)**		**Apply**
Q.6 ……	**(10 marks)**		**Apply**
SECTION – III			
Q.7……	**(10 marks)**		**Analyze**
Q.8……	**(10 marks)**		**Evaluate**
Q.9……	**(10 marks)**		**Evaluate/Create**

XVI. Programme Structure (Total Credits: 95)

SEMESTER I

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
101	Object Oriented Programming using Java	Core Courses (Theory)	3	2	1	-	50	50	100
102	Applied Database Management Systems (ORACLE)	Core Courses (Theory)	4	2	1	1	50	50	100
103	Computer Organization and Architecture	Core Courses (Theory)	3	2	1	-	50	50	100
104	Lab on Object Oriented Programming Using Java	Core Courses (Practical)	2	-	-	2	50	50	100
105	Client-Side Web Development	Core Courses (Practical)	3	2	-	1	50	50	100
106	Python Programming	Core Courses (Practical)	2	1	-	1	--	50	50
107	Soft Skills for Computing Professionals	Skill Enhancement	2	2	-	-	50	-	50
108	Organization Behaviour	Multi-disciplinary	2	2	-	-	50	-	50
109	Open Course - Any one from Bucket	Skill Enhancement	2	2	-	-	50	-	50
Total			**23**	**15**	**03**	**05**	**450**	**250**	**700**

SEMESTER II

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
201	Data Structures using Python	Core Courses (Theory)	3	2	1	-	50	50	100
202	Object-Oriented Software Engineering & Agile Practices	Core Courses (Theory)	3	2	1	-	50	50	100
203	Operating Systems for Programmers	Core Courses (Theory)	3	2	1	-	50	50	100
204	Lab on Data Structures using Python	Core Courses (Practical)	3	-	-	3	50	50	100
205	Server-Side Web Development	Core Courses (Practical	3	2	-	1	50	50	100
206	EG1-1	Elective Courses	3	2	1	-	100	-	100
207	EG2-1	Elective Courses	3	2	1	-	100	-	100
208	Minor Project - 1 *	Minor Project	3	-	-	3	-	100	100
209	Open Course -	Skill Enhancement	2	2	-	-	50	-	50
Total			**26**	**14**	**5**	**7**	**500**	**350**	**850**

SEMESTER III

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
301	Digital Transformation & Emerging Technology	Core Courses (Theory)	3	2	1	-	50	50	100
302	Software Design Principles and Design Patterns	Core Courses (Theory)	3	2	1	-	50	50	100
303	Computer Networking & cloud computing	Core Courses (Theory)	3	2	1	-	50	50	100
304	Web Application Development using .NET	Core Courses (Practical)	3	2	-	1	50	50	100
305	Generative AI & Prompt Engineering	Core Courses (Practical	3	2	-	1	50	50	100
306	EG1-2	Elective Courses	3	2	1	-	100	-	100
307	EG2-2	Elective Courses	3	2	1	-	100	-	100
308	Minor Project – 2*	Minor Project	3	-	-	3	-	100	100
309	Open Course - Any one from bucket of Open courses	Multi-disciplinary / Skill Enhancement	2	2	-	-	50	-	50
Total			**26**	**16**	**05**	**05**	**500**	**350**	**850**

* 208-Minor Project-1 and 308- Minor Project-2 courses may be completed at home practical 6 hours per week

SEMESTER IV

Course Number	Course Title	Course Type	Credits	Hours / Week			IA	UE	Total Marks
				L	T	P			
401	Seminar on Recent Trends in IT	Core Courses (Theory)	4	-	4	--	50	50	100
402	MOOC-1	MOOCs	2	-	2	-	-		
403	MOOC-2	MOOCs	2	-	2	-	-		
404	Internship - Major Project	Internship - Major Project	12			12		100	100
Total			**20**	**-**	**8**	**12**	**50**	**150**	**200**

Credits earned fr+666om MOOCs 2 + 2

Open courses Bucket

1	Cyber Security
2	Financial Literacy
3	IKS-Yoga and Wellness
4	Indian Constitution
5	Business Intelligence using PowerBI
6	Data Visualization using PowerBI
7	AI Essentials
8	Data Analytics using Python
9	Fundamentals of Google Analytics
10	Entrepreneurship Development and Start-up management
11	Universal Human Values
12	Digital Technology
13	Human Psychology at Workplace
14	Water Management & Sustainability
15	Social Change in Technology
16	Economics for IT Industry
17	Foreign Language

18	Emotional Intelligence and Well-being
19	Design Thinking and Innovation
20	Cyber Laws and Digital Ethics
21	Digital Marketing Fundamentals

XVII. List of elective (2 Year Programme)

Students must select two elective groups and complete one course from each group in Semester II, followed by the second course from those same groups in Semester III.

Course Code	Elective Group	Subject Code	Subject
EC-01	Digital Commerce	E1	Digital Business Models and E-Commerce
		E2	Customer Analytics and Digital Marketing
EC-02	Big Data	E1	Introduction to Big Data
		E2	Big Data Analytics
EC-03	AI-ML	E1	AI Foundations and Intelligent Systems
		E2	Applied Machine Learning
EC-04	Data Science	E1	Data Analytics and Visualization
		E2	Statistical Learning and Predictive Analytics
EC-05	Cloud Computing	E1	Fundamentals of Cloud Computing
		E2	Cloud Platforms and Application Deployment
EC-06	Internet of Things (IoT)	E1	Internet of Things Fundamentals and Applications
		E2	Device Programming and Application Development
EC-07	Full Stack Development	E1	API Design and Full-Stack Engineering
		E2	Microservices and Cloud-Native Applications
EC-08	Cyber Security	E1	Cyber Security and Data Protection
		E2	Security Auditing and Risk Management
EC-09	Mobile Computing	E1	Foundations of Mobile Computing
		E2	Mobile Application Development
EC-10	Blockchain Technology	E1	Fundamentals of Blockchain Technology
		E2	Blockchain Technology and Applications

XVII. Bridge Course:

(offered in Semester-1 to complete the eligibility criteria - Institutional Continuous Assessment)

- Mathematical Foundation (for students without mathematics background)
- Introduction to Programming (for students without computer science background)

Bridge Course I :

This course is designed and compulsory for the students from Non-Mathematics background and who have not completed mathematics in their 12^{th} or graduation course. The course can be conducted concurrently with semester I courses. The evaluation of this course will be at institute level for 100 marks. The student must score minimum 40 marks to pass this course. There will be no credits assigned to this Bridge Course.

Subject Name	Bridge Course I Mathematical Foundation
No. of Credits	00
Course Objective	To prepare background of the student to study courses in MCA
Cognitive Abilities	**Course Outcome as per Blooms Taxonomy**
Remembering	Remembering basic concepts and their representations
Understanding	Understanding applications of various discrete structures like sets, relations, graphs etc.
Applying	Applying various structures to represent problem data.
Analyzing	Learn to analyze the data for the given problem for representing it using proper structure.
Evaluating	Evaluate the problem for proper discrete structures.
Creating	Design new structures based on basic discrete structures to represent data..
Text Books	Discrete Structures by Kenneth Rosen
Course Plan	
Unit	**Content**
1.	**Set Theory :** Definition of a set, Representation of elements of sets, Methods of representing sets , types of sets, operations on sets , cardinality of a set, Principle of Inclusion and Exclusion , Venn Diagram , Proof by using Venn diagram
2.	**Functions and Relations :** Definition of Function, Types of Functions ,Composite Function, Relation definition, representation of relations
3.	**Logic:** Propositions, Logic Operations-Negation, Disjunction, Conjunction, Conditional and Biconditional, Truth Tables of compound propositions, Translating English sentences in to logical statements and vice versa, Logic gates and circuits
4.	**Matrices:** Matrix Definition, General Form, Representation of matrix in computers, Types of matrices, Operations on matrices: Addition, Subtraction and Multiplication, transpose , row / column transformations , Inverse of the matrix by Co-factor and Adjoint method, solutions to three variable problems by using matrices, application problems of matrices
5.	**Graphs -** Graph terminologies, types of graphs , representation of graph in computers, Paths, Eular and Hamilton graphs, graph colorings.

Bridge Course II:

This course is designed and compulsory for the students from Non-IT background. The course can be conducted concurrently with semester I courses. The evaluation of this course will be at institute level for 100 marks. The student must score minimum 40 marks to pass this course. There will be no credits assigned to this Bridge Course.

Subject Name	Bridge Course II Introduction to Programming
No. of Credits	00
Pre Requisite	Basic Mathematics and MSCIT course
Cognitive Abilities	**Course Outcome as per Blooms Taxonomy**
Remembering	Basic formula for finding areas, volumes, graphical representation of data is to be remembered.
Understanding	Do calculations by using formulas, algorithm, C program structure are to be understood
Applying	Apply basic knowledge of mathematics and computers to write programming codes.
Analyzing	Analyze the problem to represent in proper format such as graphs, trees for effective working
Evaluating	Evaluate the programs or problems for algorithms, logic
Creating	Creating proper program logic so as to reduce lines of codes is expected
Unit	**Content**
1.	Algorithm ,flow charts, integers, division, relations, relations and their types, representation of relation in computer memory, number conversion systems
2.	Trees, applications of trees, tree traversal algorithms, minimum spanning trees
3.	Fundamentals of C programming, Keywords and Identifiers, Constants, Variables, Data types, Declaration of variables, Declaration of variables as constant, Operators, Types of operators, Input and Output functions - printf(), scanf(), getchar(), putchar(), Formatted input and formatted output.
4.	Control Statements- Sequential, Selection, Iteration Statements, Branching structure- if statement, if-else statement, Nested if-else statement, else if Ladder, Conditional operator, switch statement, Loop control structures- while loop, do-while loop, for loop, Nested for loop, Jump statements-break, continue, goto statements
5.	Function call, return statement, Function parameters, Types of functions, Arrays and functions
6.	Introduction to OOP concepts.
Text Books	1.C programming by Yashwant Kanetkar
Reference Books	C Programming language by Brain W. Kernighan & Ritchie

Detail Syllabus

Sem-I

Programme: MCA			
Semester	**Course Code**	**Course Title**	
I	101	Object Oriented Programming Using Java	
Type of Course	**Credits**	**Evaluation**	**Marks**
Core Course	3	IA:UE	50:50

Course Objectives:

1. Familiarize learners with the Java feature, programming environment, tools, data types, operators and control flow.
2. To train learners in building modular programs using classes, packages, and making use of access specifiers for access controls
3. To demonstrate code reusability and runtime flexibility via inheritance, abstraction, and polymorphism.
4. To train student to make use of exception handling to prevent programs from crashing and how to read/write data using Java API.
5. Familiarise students to make use of multithreading, synchronization, and inter-thread communication
6. To equip students with concepts of generics and make use of collections types to organize, store, manipulate and find data easily.

Course Outcomes:

After successful completion of the course, students will be able to

CO1: Explain Java features, Java program structure, data types, operators and working of conditional and iterative statements.

CO2: Describe object-oriented programming principles and develop Java programs using classes, objects, constructors, packages, and access control mechanisms.

CO3: Apply inheritance, abstraction, interfaces, and polymorphism for software development.

CO4: Implement exception handling and file processing in Java applications.

CO5: Develop concurrent applications using multithreading and synchronization techniques.

CO6: Use Java Generics and Collection Framework for efficient data storage, retrieval, and manipulation.

Unit No.	Contents
Unit I	**Introduction to Java** • **Introduction to Coding**: What is programming, how Java works, and setting up the development environment.

	• **Program Structure**: Understanding the structure of a basic Java program, the main method, and printing text to the screen. • **Data Storage**: Working with variables, understanding primitive data types (integers, decimals, characters, booleans), and assigning values. • **Using Java.util.Scanner**: creating scanner to read user data and using it. • **Operators**: Arithmetic, comparison, logical and shift operators • **Decision Making**: Writing conditions using if, if-else, else-if ladders, and handling multiple choices with switch-case statements • **Loops**: Repeating code using for loops, while loops, and do-while loops; controlling loops with break and continue. • **Arrays**: Declaring and initializing One-Dimensional Arrays, accessing items by index, and using loops to read or modify array data, Working with multi-dimensional Arrays **(**Two-Dimensional Arrays and Jagged Arrays).
Unit 2	**Working with classes, Objects and packages** • **Introduction to OOP**: Thinking in terms of real-world objects: abstraction, encapsulation, the difference between a blueprint (Class) and an actual item (Object). Writing and using a first class. • **Object Initialization**: Understanding how constructors work, understanding and writing default constructors, and passing values during object creation. • **Access Modifiers and Non-Access modifiers:** Data Protection using access modifiers (public vs. private) and using getter and setter methods to safely update object variables. Making use of static and final. • **Working with Methods**: Writing custom functions, passing values into methods (parameters), and getting values back (return types). Concept of pass by value and pass by reference, static methods vs non-static methods • **String Fundamentals**: Creating text variables with String, understanding that strings cannot be changed directly (immutability), and concepts of String Pool, using methods for text/string manipulation, making use StringBuilder and StringBuffer for text/string that changes frequently. • **Package management:** understanding use packages and creating user defined packages for code management
Unit 3	**Inheritance and Polymorphism** • **Code Reuse (Inheritance):** Extending existing classes using the extends keyword, understanding the parent (superclass) and child (subclass) relationship. • **The super Keyword:** Accessing parent methods and parent constructors from inside a child class. • **Polymorphism (Many Forms):** The basic concept of performing a single action in different ways, using compile polymorphism by Method Overloading, • **Runtime Polymorphism/Flexibility:** Method Overriding (rewriting a parent's method inside a child class to change its behaviour). • **Abstract Classes and Interfaces:** Creating incomplete guide classes using abstract that serve as templates for other classes to build upon. **Interfaces:**

	Defining standard rulebooks or contracts using the interface keyword that other classes must follow, Writing and using interfaces, extending interface • **Inner classes:** Concept of inner class and using them
Unit 4	**Exception Handling and Java I/O** • **Handling Faults**: What are runtime errors (exceptions) and how do they differ from syntax errors that stop a program from compiling. Types of exceptions • **Safe Code Execution**: Protecting code from crashing using try and catch blocks; running cleanup code inside finally blocks, handling different types of errors cleanly using multiple catch blocks. • **Passing the Blame**: Using throw and throws to explicitly trigger errors or warn other parts of the program that an error might happen. • **Concept of Streams**: Understanding how data flows; differences between **Byte Streams** (for binary/raw data) and **Character Streams** (for 16-bit text data). • **The Console**: Direct interactions using `System.out` (standard output), `System.in` (standard input), and `System.err` (standard error routing). • **File Management**: Understanding the `File` class to check properties, create, delete, and look up files or directories on disk. • **Byte-Oriented Streams**: Core operations using abstract `InputStream` and `OutputStream` classes; reading and writing raw bytes using `FileInputStream` and `FileOutputStream`. • **Character-Oriented Streams**: Text processing using abstract `Reader` and `Writer` classes; reading and writing plain text characters using `FileReader` and `FileWriter`. • **Buffered Streams**: Introduction to performance buffering; managing raw primitives using `DataInputStream` and `DataOutputStream`; efficient text block reading and writing using `BufferedReader` and `BufferedWriter`. • **Saving the Object States**: Preserving object states across executions using `ObjectInputStream` and `ObjectOutputStream` for **Serialization**
Unit 5	**Introduction to Multithreading** • **Concept of Threads**: Single-threaded vs. multi-threaded tasks; processes vs. threads. • **Creating a Thread**: Using the `Thread` class and implementing the `Runnable` interface. • **Java Thread Lifecycle**: Thread states (New, Runnable, Running, Blocked, Waiting, Terminated). • **Thread Priorities**: Setting and getting thread priorities (`MIN_PRIORITY`, `NORM_PRIORITY`, `MAX_PRIORITY`). • **Thread Controls: Managing active tasks using execution states, pausing execution with Thread.sleep(), and forcing tasks to wait for others using .join().** • **Synchronization**: Avoiding race conditions; using synchronized methods and synchronized blocks.

	• **Inter-Thread Communication**: Coordinated execution using `wait()`, `notify()`, and `notifyAll()`. • **Deadlocks**: Understanding how deadlocks occur and how to avoid them.
Unit 6	**Generics and Java Collection Framework** • **Introduction to Generics**: Understanding type safety without code duplication; generic types (classes/interfaces) and generic methods. • **Collection Hierarchy**: Structural overview of the java.util framework and its core sub-interfaces. • **Lists & Sequences**: Dynamic data handling using ArrayList, legacy thread-safe Vector, and doubly-linked LinkedList implementation. • **Collection Traversal**: Accessing, reading, and removing elements sequentially using the Iterator interface. • **Set Hierarchy & Basic Sets**: Storing unique elements via the Set hierarchy; performance behaviors of HashSet and insertion-ordered LinkedHashSet. • **Object Identity in Hashing**: The fundamental role and structural contract of the equals() and hashCode() methods in avoiding bucket collisions. • **Sorted Sets & Ordering**: Sorting uniqueness using TreeSet; implementing custom object ordering using the Comparable and Comparator interfaces. • **Introduction to Map**: Key-value pairing concepts; structural layout of the Map Hierarchy. • **Map Implementations**: Managing data dictionaries using general-purpose HashMap and synchronized legacy HashTable

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
1	CO1	Lectures, Syntax Demonstrations, Interactive Discussions, Code Tracing	Remember, Understand
2	CO2	Lectures, Coding Exercises, Memory Mapping Tutorials, Laboratory Practice	Understand, Apply, Analyze
3	CO3	OOP Case Studies, Object Hierarchy Design Exercises, Programming Assignments	Apply, Analyze, Evaluate
4	CO4	Problem-Based Learning, Hands-on Crash Simulation, Using File IO classes to read and write data	Apply, Analyze, Evaluate, Create
5	CO5	Live Concurrency Coding Exercises, Multi-tasking Mini Projects, Collaborative Roleplay Learning	Apply, Analyze, Evaluate, Create
6	CO6	Collection-Based Application Development, Data Structure Case Studies, Project-Based Learning	Apply, Analyze, Evaluate, Create

Reference Books				
Sr. No.	**Name of the Author**	**Title of the Book**	**Year and Edition**	**Publisher**
1	Herbert Schildt	Java: The Complete Reference	12th Edition, 2021	McGraw-Hill Education
2	Cay S. Horstmann	Core Java Volume I: Fundamentals	13th Edition, 2024	Pearson Education
3	Paul Deitel & Harvey Deitel	Java: How to Program	12th Edition, 2022	Pearson Education
4	Kathy Sierra & Bert Bates	Head First Java	3rd Edition, 2022	O'Reilly Media
5	Joshua Bloch	Effective Java	3rd Edition, 2017	Addison-Wesley Professional

Online Resources:		
Sr. No.	**Resources**	**Website URL**
1	Oracle Java Documentation	https://docs.oracle.com/javase
2	NPTEL – Programming in Java	https://nptel.ac.in
3	SWAYAM MOOC Platform	https://swayam.gov.in
4	Spoken Tutorial (IIT Bombay)	https://spoken-tutorial.org
5	Coursera – Java Programming Specializations	https://www.coursera.org

Programme: MCA			
Semester	**Course Code**	**Course Title**	
I	102	Applied Database Management Systems	
Type of Course	**Credits**	**Evaluation**	**Marks**
Core Course	4	IA: UE	50:50

Course Objectives:

1. To provide foundational knowledge of database concepts, architecture, languages and the database development life cycle.
2. To develop skills in designing databases using data models, ER modelling and Enhanced ER concepts.
3. To impart knowledge of relational database design, integrity constraints, relational algebra and normalization.
4. To develop an understanding of transaction processing, concurrency control, recovery, security and database storage techniques.
5. To develop practical skills in creating, manipulating and querying relational databases using SQL.
6. To develop proficiency in creating and managing database objects such as views, indexes, sequences and synonyms.

Course Outcomes:

CO1: Explain fundamental database concepts, DBMS architecture, database languages and the database development life cycle.

CO2: Design conceptual database models using ER and Enhanced ER modelling techniques.

CO3: Design normalized relational schemas using integrity constraints, relational algebra and functional dependencies.

CO4: Apply transaction management, concurrency control, recovery, security, indexing and hashing techniques.

CO5: Create, manipulate and retrieve data using SQL commands, constraints, functions, joins and subqueries.

CO6: Create and manage views, indexes, sequences and synonyms for efficient database applications.

Unit No.	Contents
Unit I	**Introduction to Database Systems:** Data, Information and Knowledge, File Processing System and its Limitations, Database Concepts and Characteristics, Advantages and Applications of DBMS, Database Environment and Database Users, Role of Database Administrator (DBA), Three-Level Database Architecture, Data Independence, Database Schema and Instance, Database Languages (DDL, DML, DCL, TCL), Types of Databases, Database Development Life Cycle (DDLC)
Unit 2	**Data Modelling and Database Design:** Data Models: Hierarchical, Network, Relational and Object-Oriented Models, Entity Relationship (ER) Model, Entities, Attributes and Relationships, Cardinality and Participation Constraints, Weak Entity Sets, Enhanced ER Model, Generalisation, Specialisation, Specialisation and Aggregation, Mapping ER Model to Relational Model, Business Database Design Case Studies
Unit 3	**Relational Database Design and Normalisation:** Relational Database Concepts, Relation, Tuple, Attribute and Domain, Integrity Constraints: Entity and Referential Integrity, Relational Algebra Operations: Join Operations.

	Functional Dependencies, Normalisation, First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF)
Unit 4	**Transaction Processing and Database Administration:** Transaction Concepts and Transaction States, ACID Properties, Concurrency Control, Lock-Based Protocols, Deadlocks, Timestamp-Based Concurrency Control, Serializability Database Recovery Techniques, Log-Based Recovery, Checkpoints and Backup, Database Security, Authentication, Authorisation and Access Control File Organisation, Indexing and Hashing Techniques
Unit 5	**SQL Programming:** Introduction to SQL, SQL Data Types, SQL Environment, Data Definition Language, Constraints: Primary Key, Foreign Key, Unique, Check, Default and NOT NULL, Data Manipulation Language, Transaction Control Language, Data Query Language, SQL Operators SQL Functions, GROUP BY and HAVING Clauses, Join Operations, Set Operators, Subqueries
Unit 6	**Database Objects:** Introduction to Database Objects, Purpose and Advantages of Database Objects; **Views:** Concept and Characteristics of Views, Creating Simple and Complex Views, Retrieving and Manipulating Data through Views, Updating Views, Read-Only Views, Modifying and Dropping Views, Advantages and Applications of Views; **Indexes:** Concept and Need for Indexing, Working of Indexes, Types of Indexes—Unique, Non-Unique, Single-Column, Composite and Function-Based Indexes, Creating and Dropping Indexes, Index Selection and Impact on Query Performance; **Sequences:** Concept and Purpose of Sequences, Creating and Altering Sequences, Sequence Parameters, Dropping Sequences and Applications of Sequences; **Synonyms:** Concept and Purpose of Synonyms, Public and Private Synonyms, Creating and Using Synonyms, Modifying and Dropping Synonyms, Advantages of Synonyms in Database Security and Object Accessibility; Practical Applications and SQL Implementation of Views, Indexes, Sequences and Synonyms.

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
1	CO1	Interactive lectures, Multimedia Presentations	Remember, Understand
2	CO2	Interactive lectures, ER diagram demonstrations, mini-project discussions	Remember, Understand, Apply
3	CO2, CO3	Interactive lectures, relational schema design exercises, case studies, flipped classroom, tutorial sessions	Remember, Understand, Apply
4	CO4	Interactive lectures, simulation of transaction processing and concurrency scenarios, case studies	Remember, Understand, Apply
5	CO5	Interactive lectures, Live SQL demonstrations, Hands-on laboratory sessions, Query-writing practice	Understand, Apply, Evaluate, Create
6	CO6	Interactive lectures, Live SQL demonstrations, Hands-on laboratory sessions, Object Creation & Query-writing practice	Understand, Apply, Evaluate, Create

Reference Books				
Sr. No.	**Name of the Author**	**Title of the Book**	**Year and Edition**	**Publisher**
1	Ramez Elmasri, Shamkant B. Navathe	Fundamentals of Database Systems	5th Edition	PEARSON Education
2	Silberschatz, H Korth, S Sudarshan	Database System and Concepts	4th Edition	McGraw-Hill.
3	Ivan Bayross	SQL, PL/SQL The Programming Language of Oracle	5th Edition	BPB Publication

Online Resources:		
Sr. No.	**Resources**	**Website URL**
1	w3schools	https://www.w3schools.com/sql/
2	Tutorialspoint	https://www.tutorialspoint.com/sql/index.htm 3
3	Javatpoint	https://www.javatpoint.com/sql-tutorial

Programme MCA			
Semester	**Course Code**	**Course Title**	
I		Computer Organization and Architecture	
Type of Course	**Credits**	**Evaluation**	**Marks**
Core Course	3	IA : UA	50:50

Course Objectives:

1. Provide a strong foundation in digital logic fundamentals, number systems, and combinational and sequential circuits used in computer organization.
2. Introduce the concepts of processor organization, instruction execution, register organization, and control unit design.
3. Explain computer arithmetic, including data representation, ALU operations, multiplication, and division algorithms.
4. Develop an understanding of memory organization, memory hierarchy, cache memory, virtual memory, and storage systems.
5. Familiarize students with input/output organization, interrupt handling, DMA, and communication interfaces used in computer systems.
6. Expose students to modern processor architectures, including RISC, CISC, multi-core, SIMD, and MIMD architectures, and their applications.

Course Outcomes: After successful completion of the course, students will be able to:

CO1. Explain the fundamentals of digital logic, number systems, code conversions, and combinational and sequential circuits used in computer systems

CO2. Analyze processor organization, instruction formats, instruction execution, addressing modes, register organization, and control unit operations

CO3. Apply integer and floating-point representations, ALU operations, and arithmetic algorithms to solve binary arithmetic problems.

CO4. Explain the organization and operation of memory hierarchy, cache memory, main memory, ROM, RAM, and virtual memory.

CO5. Describe input/output organization, interrupt handling, DMA, bus organization, communication interfaces, and secondary storage devices

CO6. Compare RISC, CISC, multi-core, SIMD, and MIMD architectures and evaluate their performance for different computing applications

Unit No.	Contents
Unit I	**Digital Logic Fundamentals: (8 Hours)** Block diagram of computer, Digital Systems and Binary Logic, Number Systems and Code Conversions, Logic Gates, Combinational Circuits: Half Adder and Full Adder, Adder-Subtractor, Incrementer, Multiplexer (MUX) and Demultiplexer (DEMUX), Encoder and Decoder

Unit 2	**Sequential Circuits (7 Hours)** Flip-Flops (SR, JK, D, T), Registers and Register Transfer, Shift Registers, Counters (Synchronous and Asynchronous), Register Transfer Language (RTL), Bus Organization, Arithmetic, Logic and Shift Microoperations
Unit 3	**Processor Organization and Instruction Execution: (8 Hours)** Von Neumann and Harvard Architectures, CPU Organization, Register Organization, Instruction Formats, Instruction Cycle, Addressing Modes, Stack Organization, Subroutine Call and Return Mechanism, Hardwired Control Unit, Microprogrammed Control Unit
Unit 4	**Computer Arithmetic (7 Hours)** Integer Representation, Signed Number Representation , Floating Point Representation (IEEE-754), Binary Arithmetic Operations Arithmetic Logic Unit (ALU), Booth's Multiplication Algorithm Division Algorithms, Arithmetic Processor
Unit 5	**Memory Organization (6 Hours)** Memory Hierarchy, Cache Memory, Cache Mapping Techniques Main Memory, ROM and RAM, Virtual Memory
Unit 6	**Input/Output Organization (9 Hours)** Input/Output Interface, Interrupt Driven I/O, Direct Memory Access (DMA), Interrupt Priority, Handshaking, Serial and Parallel Communication, USB and PCI Bus, Secondary Storage Devices RISC Architecture, CISC Architecture, Multi-core Processors, SIMD and MIMD Architectures

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit	CO Mapping	Pedagogy	Bloom's Cognitive Level
Unit I	CO1	Lectures, Demonstration, Problem Solving, Laboratory Simulation	Understand (L2), Apply (L3)
Unit II	CO2	Interactive Lectures, Case Studies, CPU Simulation	Understand (L2), Analyze (L4)
Unit III	CO3	Numerical Exercises, Problem Solving	Apply (L3), Analyze (L4)
Unit IV	CO4	Lectures, Memory Simulation, Group Discussion	Analyze (L4), Evaluate (L5)
Unit V	CO5	Demonstration, Case Study, Laboratory Activities	Understand (L2), Apply (L3)
Unit VI	CO6	Seminar, Research Paper Discussion, ICT-based Learning	Analyze (L4), Evaluate (L5)

Reference Books				
Sr. No.	**Author**	**Title**	**Edition**	**Publisher**
1	William Stallings	*Computer Organization and Architecture: Designing for Performance*	11th Edition	Pearson
2	Carl Hamacher, Zvonko Vranesic & Safwat Zaky	*Computer Organization*	6th Edition	McGraw-Hill
3	M. Morris Mano	*Computer System Architecture*	3rd Edition	Pearson
4	R. P. Jain	*Modern Digital Electronics*	5th Edition	McGraw-Hill Education
5	Floyd	*Digital Fundamentals*	11th Edition	Pearson

Online Resources:		
Sr. No.	**Resources**	**Website URL**
1	NPTEL – Computer Organization and Architecture	https://nptel.ac.in
2	MIT OpenCourseWare	https://ocw.mit.edu
3	GeeksforGeeks – Computer Organization	https://www.geeksforgeeks.org
4	TutorialsPoint – Computer Organization	https://www.tutorialspoint.com

Programme: MCA			
Semester	**Course Code**	**Course Title**	
I	**104**	**Lab on Object Oriented Programming Using Java**	
Type of Course	**Credits**	**Evaluation**	**Marks**
Core Course (Practical)	**3**	**IA:UE**	**50:50**

Course Objectives:

1. Provide hands-on experience in Java programming environment and basic programming constructs.
2. Develop practical skills in designing and implementing classes, objects, packages, and methods.
3. Enable students to implement inheritance, interfaces, abstraction, and polymorphism through Java programs.
4. Train students to handle exceptions and perform file-based input/output operations.
5. Develop multithreaded applications using synchronization and inter-thread communication.
6. Apply Generics and Collection Framework for solving real-world programming problems.

Course Outcomes:

After successful completion of the course, students will be able to

CO1: Develop Java programs using variables, operators, control structures, and arrays.

CO2: Design and implement object-oriented applications using classes, objects, constructors, methods, and packages.

CO3: Build reusable software components using inheritance, interfaces, abstraction, and polymorphism.

CO4: Create robust applications using exception handling and file processing techniques.

CO5: Develop concurrent Java applications using multithreading and synchronization.

CO6: Implement applications using Generics and Java Collection Framework.

Unit No.	Contents
Unit I	**Introduction to Java** 1. **Use** Java development tools to create and execute basic programs. 2. **Apply** command-line arguments and Scanner class for input processing. 3. **Use** arithmetic and relational operators in problem-solving applications. 4. **Analyze** expressions involving logical and bitwise operators. 5. **Apply** conditional statements to implement decision-making logic. 6. **Design** menu-driven applications using switch-case constructs. 7. **Develop** programs using iterative control structures. 8. **Implement** applications using one-dimensional arrays. 9. **Apply** two-dimensional arrays for matrix operations. 10. **Analyze** and manipulate data using jagged arrays
Unit 2	**Working with classes, Objects and packages** 1. **Create** classes and objects to model real-world entities. 2. **Implement** object initialization using constructors. 3. **Apply** constructor overloading in object-oriented applications.

	4. **Use** access modifiers to achieve encapsulation. 5. **Design** applications using getter and setter methods. 6. **Develop** programs using static members and methods. 7. **Apply** the final keyword for constants and restricted inheritance. 8. **Implement** method overloading and parameter passing techniques. 9. **Use** String class methods for text processing. 10. **Analyze** performance differences between String, StringBuilder, and StringBuffer 11. **Create and use** user-defined packages for modular programming.
Unit 3	**Inheritance and Polymorphism** **1. Implement** single inheritance for code reuse. **2. Implement** multilevel inheritance hierarchies. **3. Design** hierarchical inheritance models. **4. Use** the super keyword to access parent class members. **5. Apply** method overloading to achieve compile-time polymorphism. **6. Implement** method overriding for runtime polymorphism. **7. Analyze** runtime polymorphic behavior in class hierarchies **8. Develop** applications using abstract classes. **9. Design** solutions using interfaces and interface inheritance. **10. Implement** inner classes for encapsulated functionality.
Unit 4	**Exception Handling and Java I/O** **1. Identify and handle** predefined exceptions in Java programs. **2. Apply** try-catch-finally blocks for reliable execution. **3. Implement** multiple catch blocks for different exception types. **4. Use** throw and throws for custom exception handling. **5. Manage** files and directories using the File class. **6. Develop** applications using FileInputStream and FileOutputStream. **7. Implement** text file processing using FileReader and FileWriter. **8. Apply** buffered streams for efficient file processing. 9. **Design** applications using object serialization and deserialization.
Unit 5	**Introduction to Multithreading** 1. **Create** threads using the Thread class. 2. **Develop** multithreaded applications using the Runnable interface. 3. **Analyze** thread lifecycle states and priorities. 4. **Apply** sleep() and join() methods to control thread execution. 5. **Implement** synchronized methods for thread safety. 6. **Design** synchronized blocks for controlled resource sharing. 7. **Develop** applications using inter-thread communication mechanisms.
Unit 6	**Generics and Java Collection Framework** 1. **Implement** generic classes and generic methods for type-safe programming. 2. **Use** ArrayList for dynamic data storage and manipulation. 3. **Compare and analyze** ArrayList, Vector, and LinkedList implementations. 4. **Apply** Iterator for traversing and modifying collection elements.

	5. **Implement** HashSet and LinkedHashSet to manage unique elements. 6. **Analyze** the role of equals() and hashCode() in collection behavior. 7. **Design** applications using TreeSet, Comparable, and Comparator for sorting custom objects. 8. **Develop** key-value-based applications using HashMap, Hashtable, and Arrays utility methods.

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
1	**CO1**	Demonstration, Guided Programming Exercises, Code Tracing, Hands-on Practice, Problem Solving Sessions	**Remember, Understand, Apply, Analyze**
2	**CO2**	Object Modeling Exercises, Laboratory Practice, Pair Programming, Case-Based Learning, Coding Assignments	**Understand, Apply, Analyze, Create**
3	**CO3**	OOP Case Studies, UML/Class Design Activities, Programming Assignments, Application Development Exercises	**Apply, Analyze, Evaluate, Create**
4	**CO4**	Problem-Based Learning, Debugging Exercises, File Handling Activities, Hands-on Error Handling Experiments	**Apply, Analyze, Evaluate, Create**
5	**CO5**	Live Coding Demonstrations, Concurrent Programming Exercises, Collaborative Learning, Mini Application Development	**Apply, Analyze, Evaluate, Create**
6	**CO6**	Collection-Based Application Development, Data Structure Case Studies, Coding Challenges, Project-Based Learning	**Apply, Analyze, Evaluate, Create**

Reference Books

Sr. No.	Name of the Author	Title of the Book	Year and Edition	Publisher
1	Herbert Schildt	Java: The Complete Reference	12th Edition, 2021	McGraw-Hill Education
2	Cay S. Horstmann	Core Java Volume I: Fundamentals	13th Edition, 2024	Pearson Education
3	Paul Deitel & Harvey Deitel	Java: How to Program	12th Edition, 2022	Pearson Education
4	Kathy Sierra & Bert Bates	Head First Java	3rd Edition, 2022	O'Reilly Media
5	Joshua Bloch	Effective Java	3rd Edition, 2017	Addison-Wesley Professional

Online Resources:		
Sr. No.	**Resources**	**Website URL**
1	Oracle Java Documentation	https://docs.oracle.com/javase
2	NPTEL – Programming in Java	https://nptel.ac.in
3	SWAYAM MOOC Platform	https://swayam.gov.in
4	Spoken Tutorial (IIT Bombay)	https://spoken-tutorial.org
5	Coursera – Java Programming Specializations	https://www.coursera.org

Programme : MCA			
Semester	**Course Code**	**Course Title**	
I	105	Client-Side Web Development	
Type of Course	**Credits**	**Evaluation**	**Marks**
Core Course (Practical)	3	IA:UE	50:50

Course Objectives:

1. **Demonstrate** the core principles of web technologies, client-server architecture, and web security models.
2. **Train** students to design structured, accessible, and responsive web layouts using modern HTML5 and CSS3.
3. **Facilitate** hands-on experience in building interactive client-side applications via JavaScript and DOM manipulation.
4. **Introduce** component-based UI development and state management patterns using the React.js framework.

Course Outcomes:

After successful completion of the course, students will be able to

CO1: Explain web architecture, browser functionality, and basic web security concept
CO2: Develop structured and responsive web pages using HTML5 and CSS3
CO3: Create dynamic and interactive web applications using JavaScript and DOM manipulation
CO4: Develop reusable user interface components and single-page applications using React.js

Unit No.	Contents
Unit I	**Web Fundamentals, HTML5 & Web Forms(09)** Internet and World Wide Web, Client–Server Architecture, HTTP and HTTPS, Browser Rendering Process, Basic Web Security Concepts, Introduction to HTML5, Document Structure and Semantic Elements, Headings, Paragraphs, Lists, Tables, Hyperlinks, Images and Multimedia, HTML5 Forms, Input Controls and Form Elements, Built-in HTML5 Validation **Practical(7 Hrs)** • Browser Developer Tools • Creating Web Pages using HTML5 • Semantic Page Design • Registration, Login and Feedback Forms • Form Validation using HTML5 Attributes
Unit 2	**CSS3 & Responsive Web Design(06)** CSS Syntax and Selectors, Colors, Fonts and Typography, Box Model Margin, Padding and Borders, Layout Design, Flexbox Responsive Design Concepts, Styling Forms and User Interfaces **Practical(6 hrs)**

	• Applying CSS to HTML Pages • Creating Responsive Layouts • Flexbox-based Page Design • Styling Forms and Navigation Menus
Unit 3	**JavaScript Programming and DOM Manipulation(08)** Introduction to JavaScript, Variables and Data Types, Operators and Expressions, Conditional Statements, Looping Statements, Arrays and Functions, DOM Structure, Selecting and Modifying Elements, Event Handling, Client-side Form Validation Practical (8 hrs) • JavaScript Programming Exercises • Calculator Application • Dynamic Content Manipulation • Form Validation Techniques • To-do List Application • Event-driven Web Applications
Unit 4	**React.js Fundamentals and Component Development (07)** Introduction to React , Virtual DOM, JSX, Functional Components, Props, State Management using useState, Event Handling in React, Component Reusability **Practical (9 hrs)** • React Environment Setup using Vite • Creating Functional Components • Managing Component State • Handling User Input • Reusable UI Components • Mini Project: React-based Interactive Web Application

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
Unit I	CO1	Lectures, Demonstrations, Interactive Discussions	Remember, Understand
Unit II	CO2	Hands-on Practice, Tutorials, Design Exercises	Understand, Apply, Analyze
Unit III	CO3	Coding Exercises, Problem Solving, Laboratory Activities	Apply, Analyze, Evaluate
Unit IV	CO4	Project-Based Learning, Mini Projects, Application Development	Apply, Analyze, Evaluate, L6 Create

Reference Books

Sr. No.	Name of the Author	Title of the Book	Year and Edition	Publisher
1	Achyut S. Godbole & Atul Kahate	*Web Technologies*	2017, 3rd Edition	McGraw-Hill Education (India)
2	David Flanagan	*JavaScript: The Definitive Guide*	2020, 7th Edition	O'Reilly Media
3	Alex Banks & Eve Porcello	*Learning React: Modern Patterns for Developing React Apps*	2024, 3rd Edition	O'Reilly Media
4	Jon Duckett	*HTML and CSS: Design and Build Websites*	2011, 1st Edition	Wiley

Online Resources:		
Sr. No.	**Resources**	**Website URL**
1	MDN Web Docs (HTML5, CSS3, JavaScript)	https://developer.mozilla.org
2	JavaScript.info (JavaScript & DOM)	https://javascript.info
3	React Official Documentation	https://react.dev
4	NPTEL – Web Technologies	https://nptel.ac.in
5	Spoken Tutorial (IIT Bombay)	https://spoken-tutorial.org

Programme: MCA			
Semester	**Course Code**	**Course Title**	
I	106	Python Programming	
Type of Course	**Credits**	**Evaluation**	**Marks**
Core Course (Practical)	02	IA	50

Course Objectives:

1. **Deliver** comprehensive instruction on Python syntax, standard input/output operations, and conditional control flow logic
2. **Train** students to efficiently select, utilize, and manipulate built-in sequential data containers, including strings, lists, tuples, and dictionaries
3. **Facilitate** hands-on mastery of functional programming techniques, modular code architectures, and scope management utilizing standard libraries.
4. **Guide** students in designing resilient applications by incorporating persistent file storage and robust exception-handling mechanisms.

Course Outcomes:

After completing the course, the students shall be able to

CO 1: Formulate imperative logic and execute structured Python programs using proper variable scopes and data types.

CO 2: Implement complex data manipulation tasks using advanced sequence operations, comprehensions, and key-value mapping.

CO 3: Deconstruct monolithic scripts into modular components using user-defined functions, arguments, and custom library packages.

CO 4: Build safe, fault-tolerant Python scripts that process external text files, log events, and isolate system runtime errors.

Unit No.	Contents
Unit I	**Foundations of Python and Control Structures** Introduction to Python, Features and Applications, Installation and Execution Modes, Python IDEs, Variables and Identifiers, Keywords, Data Types, Input and Output Functions, Operators and Expressions, Type Conversion, Comments and Documentation Strings, Decision Making (if, if-else, nested if, elif), Iterative Statements (while, for, nested loops), Control Statements (break, continue, pass).
Unit 2	**Functions and Modular Programming** Functions, Defining and Calling Functions, Function Arguments, Default Arguments, Keyword Arguments, Variable-Length Arguments, Return Statement, Recursive Functions, Scope and Lifetime of Variables, Lambda Functions, Modules, Creating User-Defined Modules, Importing Modules, Introduction to Packages.
Unit 3	**Strings and Python Collections** Strings, String Operations, Slicing, String Methods, String Formatting, Lists, List Operations, List Methods, List Comprehension, Tuples, Sets, Set Operations, Set Comprehension, Dictionaries, Dictionary Operations, Dictionary Comprehension, Nested Collections.
Unit 4	**Exception Handling, File Processing and Standard Libraries**

	Exception Handling (try, except, else, finally, raise), File Handling (open, close, read, write, append), File Methods, Standard Libraries (math, random, datetime, os), Introduction to Command-Line Arguments.

Unit–CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
1	CO1	Lectures, Demonstrations, Coding Exercises, Problem Solving	Remember, Understand, Apply
2	CO2	Tutorials, Programming Assignments, Hands-on Practice	Apply, Analyze
3	CO3	Laboratory Exercises, Case Studies, Problem Solving	Apply, Analyze, Evaluate
4	CO4	Project-Based Learning, Programming Assignments, Mini Applications	Apply Analyze, Evaluate, Create

Reference Books

Sr. No.	Name of the Author	Title of the Book	Year and Edition	Publisher
1	Allen B. Downey	*Think Python: How to Think Like a Computer Scientist*	2nd Edition, 2016	O'Reilly Media
2	Reema Thareja	*Python Programming: Using Problem Solving Approach*	2nd Edition, 2019	Oxford University Press
3	Eric Matthes	*Python Crash Course*	3rd Edition, 2023	No Starch Press
4	Charles Dierbach	*Introduction to Computer Science Using Python*	2nd Edition, 2015	Wiley
5	Dr. U. Dinesh Kumar	*Python Programming*	2021 Edition	Pearson India

Online Resources:

Sr. No.	Resources	Website URL
1	Python Tutorial	https://docs.python.org/3/tutorial
2	NPTEL – Python Programming Courses	https://nptel.ac.in
3	SWAYAM MOOC Platform	https://swayam.gov.in
3	Spoken Tutorial (IIT Bombay) – Python	https://spoken-tutorial.org

Programme: MCA			
Semester	**Course Code**	**Course Title**	
I	107	Soft Skills for Computing Professionals	
Type of Course	**Credits**	**Evaluation**	**Marks**
Skill Enhancement	2	IA	50

Course Objectives:

1. Develop effective communication skills for professional interactions through oral, written, and digital communication in computing environments.
2. Enhance teamwork, leadership, emotional intelligence, and interpersonal skills required for effective collaboration in software development and professional workplaces.
3. Cultivate critical thinking, problem-solving, ethical reasoning, and professional conduct for addressing workplace challenges in computing environments.
4. Build career readiness through resume preparation, interview skills, professional networking, personal branding, and responsible use of AI-powered productivity tools.

Course Outcomes:

After successful completion of the course, students will be able to

CO1: Communicate effectively in technical and professional environments using oral, written, and digital communication.

CO2: Collaborate effectively in multidisciplinary software development teams by demonstrating teamwork, leadership, emotional intelligence, and conflict resolution skills.

CO3: Apply critical thinking, ethical reasoning, and professional conduct while solving workplace problems in computing environments.

CO4: Demonstrate career readiness through resume preparation, interview skills, professional networking, and responsible use of AI tools.

Unit	Contents

1	**Professional Communication for Computing**
	• Communication process and barriers
	• Active listening and questioning techniques
	• Verbal and non-verbal communication
	• Professional email and chat etiquette
	• Technical writing
	  o Bug reports
	  o Meeting minutes
	  o Software documentation
	• Presentation skills for technical audiences
	• Cross-cultural communication in global software teams

2	**Teamwork and Workplace Effectiveness** • Team dynamics in Agile environments • Collaboration and accountability • Emotional Intelligence (EQ) • Conflict management and negotiation • Time management and productivity • Critical thinking and creative problem solving • Adaptability and continuous learning
3	**Professionalism, Ethics and Career Readiness**

	• Professional ethics for software professionals • Workplace etiquette and organizational behaviour • Digital citizenship and online professionalism • Responsible use of AI tools (prompting, verification, privacy, plagiarism) • Resume and LinkedIn profile preparation • Interview skills (HR and technical) • Personal branding and networking
4	**Capstone Professional Skills Workshop (6 Hours)** Integrated practice through: • Group presentation • Software project pitch • Case studies • Role plays • Peer feedback • Reflection and self-assessment

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
1	CO1	Interactive Lectures, Technical Presentations, Email Writing Exercises, Role Plays, Peer Feedback	Understand, Apply
2	CO2, CO3	Scrum Simulation, Group Discussions, Team Problem-Solving Challenges, Reflection Journal	Apply, Analyze, Evaluate
3	CO1, CO2, CO3, CO4	Case Studies, Mock Interviews, Resume Review, LinkedIn Profile Creation, AI-Assisted Communication Exercises	Apply, Analyze, Evaluate, Create
4	CO1, CO2, CO3, CO4	Software Project Pitch, Group Presentation, Role Plays, Peer Assessment, Capstone activity	Apply, Analyze, Evaluate, Create

Reference Books				
Sr. No.	Name of the Author	Title of the Book	Year and Edition	Publisher
1	Alex, K.	Soft Skills: Know Yourself and Know the World	2022, 4th Edition	S. Chand Publishing
2	Raman, Meenakshi & Sharma, Sangeeta	Communication Skills	2021, 3rd Edition	Oxford University Press
3	Goleman, Daniel	Emotional Intelligence: Why It Can Matter More Than IQ	2006, 10th Anniversary Edition	Bantam Books
4	Covey, Stephen R.	The 7 Habits of Highly Effective People	2020, 30th Anniversary Edition	Simon & Schuster
5	Carnegie, Dale	How to Win Friends and Influence People	2010, Revised Edition	Simon & Schuster
6	De Bono, Edward	Six Thinking Hats	2017, Revised Edition	Penguin Books

Online Resources:

Sr. No.	Resources	Website URL
1	MIT OpenCourseWare – Communication and Professional Skills Resources	https://ocw.mit.edu
2	Coursera – Communication, Teamwork and Career Development Courses	https://www.coursera.org
3	LinkedIn Learning – Professional Communication and Career Skills	https://www.linkedin.com/learning
4	Google for Developers – Technical Writing and Communication Resources	https://developers.google.com/tech-writing
5	UNESCO – Guidance for Generative AI in Education and Research	https://unesdoc.unesco.org

Programme: MCA			
Semester	**Course Code**	**Course Title**	
I	108	Organizational Behaviour	
Type of Course	**Credits**	**Evaluation**	**Marks**
Multi-disciplinary	02	IA	50

Course Objectives

- **Provide** a comprehensive analysis of individual, group, and organizational behavior dynamics unique to technology-driven workplaces.
- **Cultivate** essential professional skills required for effective teamwork, strategic communication, and leadership within IT organizations.
- **Analyze** the impacts of motivation strategies, organizational culture, and change management frameworks within software and digital enterprises.]
- **Train** students to apply organizational behavior concepts to systematically optimize productivity, foster corporate innovation, and enhance employee well-being.

Course Outcomes

After completing the course, the students shall be able to

CO1- Explain key concepts of organizational behavior and their relevance to IT organizations.

CO2- Analyze individual and team behavior in software development and project environments.

CO3- Apply leadership, motivation, and communication techniques to improve organizational effectiveness.

CO4- Evaluate organizational culture, change management, and emerging workplace trends in the IT industry

Unit No.	Contents
Unit I	Introduction to Organizational Behavior; OB models; Individual behavior; Personality-Determinants and Theories, values- types of Values, attitudes-Formation of attitude;
Unit 2	Perception-Factors affecting Perception, Perceptual Process, Perceptual Error, Learning-Process of learning and developing learning organisation; Motivation-Process, Theories (Maslow, Herzberg, Vroom); Employee engagement; Managing knowledge workers and software professionals.
Unit 3	Leadership- Types, Leadership theories; Leadership in technology organizations; Power and politics; Decision-making; Emotional intelligence; Diversity and inclusion in IT workplaces, Work ethics.
Unit 4	Group-Development process, dynamics; Team building; Agile and Scrum teams; Managing virtual and hybrid teams; Conflict and negotiation in software projects, Organizational culture; Change management; Future of work

CO Mapping, Pedagogy and Bloom's Cognitive Levels

Unit No.	CO Mapping	Pedagogy	Bloom's Cognitive Levels
1	CO1	Lectures, Case Studies	Understand, analyze
2	CO2	Flipped classroom, Discussion, Case Studies	Apply,
3	CO3	Flipped classroom, Case Study, Discussion	Evaluate, create
4	CO4	Flipped classroom, role Play, case Study	Analyze, evaluate

Reference Books				
Sr. No.	**Name of the Author**	**Title of the Book**	**Year and Edition**	**Publisher**
01	Stephen P. Robbins & Timothy A. Judge	Organizational Behavior	2022, 18th	Pearson
02	Parikh & Gupta	Organizational Behavior	2021, Latest	McGraw-Hill
03	Fred Luthans	Organizational Behavior	2023, 14th	McGraw-Hill

Online Resources:		
Sr. No.	**Resources**	**Website URL**
1	SHRM	https://www.shrm.org
2	Gallup	https://www.gallup.com/workplace
3	Harvard Business Review	https://hbr.org/topic/organizational-behavior

MOOCs:		
Sr. No.	**Resources**	**Website URL**
1	Coursera	https://www.coursera.org
2	EDX	https://www.edx.org
3	NPTEL	https://onlinecourses.nptel.ac.in